© 1994 Balkema, Rotterdam, ISBN 90 5410 389 2

Modern computer technologies in hydrologic data management

N. Papakostas
Division of Computer Engineering, National Technical University of Athens, Greece
I. Nalbantis & D. Koutsoyiannis
Division of Water Resources, Hydraulic and Maritime Engineering, National Technical University of Athens, Greece

ABSTRACT: Advanced water resources management is facilitated by hydrologic data management using modern computer technologies. By means of such technologies, a distributed data management system for hydrologic, meteorologic and hydrogeologic historical information has been built. In this paper, we present the fundamental design and implementation principles and features of the system. Information classification, local and distributed database design and network architecture are described.

INTRODUCTION

During the past three decades, the evolution of computer technologies has changed the way hydrologic and meteorologic data are stored, processed and managed. Several systems that employ such technologies have been built to cover a wide spectrum of methods, tools and requirements. First generation systems, based on central mainframe computers and programming languages like FORTRAN, have pioneered the area of electronic hydrologic and meteorologic data storage and handling. Such systems include NAQUADAT in Canada and WATSTORE in the USA (Rodda & Flanders, 1985). The second generation systems, introducing personal computers and database technologies have emerged during the 1980's. Such systems include HYMOS in the Netherlands (Ogink 1981), HYDATA in Great Britain (Institute of Hydrology, 1991) and HYDRA-PC in Greece (Koutsoyiannis et al., 1991).

Third generation systems, involving relational database technologies, distributed systems and graphics are now being developed. Examples of such systems are (Nalbantis et al. 1992): Sequoia 2000 in the USA (Stonebraker, 1992), National Water Information System II in the USA (USGS, 1991) and Compu-Mod in Canada (Environment Canada, 1992).

Greece has been mentioned by WMO (1977) as one of the countries that use computer technologies to store hydrometeorologic data. It had lacked, however, a modern, third generation large scale system capable of handling the complex and diverse requirements of the various services that collect and store hydrometeorologic data. Such a system named HYDROSCOPE is now under development.

MAJOR DESIGN GOALS

HYDROSCOPE is created as a joint research project (Tolikas et al., 1993). Fourteen public services participate including ministries, universities and research centres, NTUA being the main partner (Table 1). The project is sponsored by the EC. It aims at the organisation, systemisation and standardisation of hydrologic, meteorologic and hydrogeologic information in Greece. Thus, its primary goal is the creation of a national databank for such information, utilising the capabilities offered by available modern computer technologies. A second goal of major importance is the creation of the framework, within which the participating organisations will be able to co-operatively store, access and process data in a uniform and consistent way. The main principles of the system are autonomy and interchange. The autonomy ensures the independence of each partner in ownership, storage and administration of its own data, according to its specific research and operational needs. The interchange does not apply only to data, but to expertise and know-how as well.

Table 1: HYDROSCOPE participating organisations

Abbreviation	Organisation
NTUA	National Technical University of Athens (Division of Water Resources, Hydraulic and Maritime Engineering)
NMS	National Meteorological Service
MA	Ministry of Agriculture
MEPPW	Ministry of Environment, Physical Planning and Public Works
NOA	National Observatory of Athens
PPC	Public Power Corporation
AUT/ DHEE	Aristotle University of Thessaloniki (Division of Hydraulic and Environmental Engineering)
AUT/ DE	Aristotle University of Thessaloniki (Division of Energy)
NRCPS "D"	National Research Centre for Physical Sciences "Demokritos"
WSSCA	Water Supply and Sewage Corporation of Athens
UA	University of Athens (Department of Applied Physics)
MIET	Ministry of Industry, Energy and Technology
CRES	Centre for Renewable Energy Sources
HALDLG	Hellenic Agency for Local Development and Local Government

AVAILABLE MODERN COMPUTER TECHNOLOGIES

Several currently available modern computer technologies have been employed, which are capable of supporting both present and future (within reasonable expectations) design goals. They are also commercially available, mature and proven enough not to put at risk the success and viability of the project. Such technologies include:

1. *Relational databases* (Codd, 1970). The state-of-the-art in commercially available database technology offers extensive data handling capabilities and a rich set of support and development tools, well suited for hydrologic applications.

2. *Distributed and decentralised databases*. By distributing computing power and data storage to many locations, as opposed to a huge central monolithic facility, site autonomy and independence, as well as enhanced flexibility and reliability are achieved.

3. *Client-server application architecture*. Data handling and application (front-end) programmes may run on different computers and communicate transparently over a network.

4. *High speed wide-area networks (WAN)*. These are composed of digital dedicated circuits and multiprotocol routers.

5. *Powerful workstations*. Downsizing computing power to the desktop level, while using leading-edge processor, disk and graphics technology results in substantial improvements of the cost/performance ratio.

6. *Easy-to-use graphical user interfaces*. These tools enable easier and friendlier user interaction.

7. *Fourth generation programming languages and object oriented techniques and tools*. Tightly coupled to the data handling facilities, they offer an integrated modern programming environment.

SYSTEM DESCRIPTION AND CURRENT STATE

HYDROSCOPE is a distributed system, currently consisting of 12 nodes. Each node is an Ethernet local area network (LAN) residing at the headquarters of the respective participating organisation (Figure 1). On the LAN, several personal computers (PCs) running MS Windows are attached. These computers are used for data entry and can also run (although slower) HYDROSCOPE's applications. The main component of the LAN is a Hewlett-Packard 9000/s700 UNIX RISC workstation running the INGRES relational distributed database management system and application development tools (INGRES Windows/4GL, etc.), and the HYDROSCOPE applications. The workstation is the database server for the node, holding any data that reside on it, as well as participating in the distributed database. The SQL language (ISO, 1986) is used for data manipulation.

The nodes themselves are interconnected via a wide area network. As most of the nodes are located in the Athens metropolitan area, dedicated leased circuits are used to link them and form a private HYDROSCOPE WAN. A multiprotocol Network Systems router at each node co-ordinates incoming/outgoing traffic for that node. Nodes that reside outside Athens are currently linked to the rest of the network via a public WAN.

The current state of the system is as follows: the WAN is set up and running, although some tests have still to be performed. All node LANs are also complete. Local databases are set up. The distributed database, to which a connection can be made from any node, is also ready. A few components of the database design have still to be implemented. A small set of real data has been inserted in the system for evaluation, debugging, verification and optimisation of both the database and the applications. The application programmes are complete to a great extent, and the system is in its final stages of implementation and testing. However, the majority of data has not been entered into the database; this task has been scheduled for the next phase of the project.

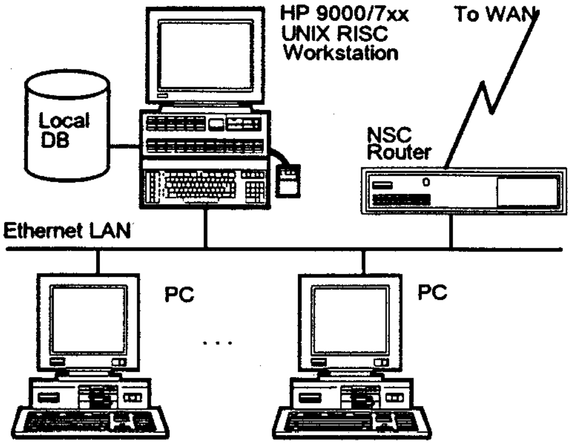

Figure 1: Typical HYDROSCOPE node architecture

CLASSIFICATION OF INFORMATION

There are three main categories of the information stored in the database (Papakostas, 1993). These are:

1. *Application information.* This information has nothing to do with the actual data. It is there just because it is required by the applications. Examples of such information are the topographic maps, the user groups, etc.

2. *Metadata or administrative information* (Figure 2). This is the information about the objects through which the actual data (see below) are located and accessed. These objects are:

Measuring stations. Each station belongs to a single service and may contain several instruments.

Instruments or measuring devices. Each instrument performs a measurement of a single hydrometeorologic variable at various time intervals, e.g. every hour, day, month or even irregularly, and thus "produces" time series data.

Time series characteristics for time series data collected by the instruments. For example, time resolution of the time series, periods with missing values, etc.

Constants, i.e. the characteristics of static data (see below, item 3) about other objects (stations, instruments, time series).

Events about stations, instruments, time series characteristics and constants. Events are chronological information about everything "important" that happens during the object's lifetime, along with an appropriate report. Examples of events are the termination of operation of a station, the repair of an instrument, the infilling of a time series, the creation of a constant, etc.

3. *Actual data* (Figure 2). This is the core of the database, the actual hydrometeorologic information stored. Data can be further classified into two subcategories:

Static (or constant) data are information about stations, instruments and time series that is time invariant or changes infrequently. Examples of such data are the lithological section or drilling information of a hydrogeologic station or the cross-section of a river. They are indexed by the station, instrument or time series to which they refer and are accessed through the constants metadata. No further processing takes place on them.

Time series data are collected from measuring devices or produced by aggregation, processing or transformations on other time series data. They are indexed by the instrument (and date) to which they refer and are accessed through the time series characteristics metadata.

The different types of the Data and Metadata objects are classified into various categories as shown in Figure 2. Stations can be of two types, primary and non-primary. The latter are of low interest as they are used rarely. If a station is non-primary, then the corresponding instruments, time series, constants and events are also considered as non-primary. Instruments are also divided in two categories: real and derived ones (Figure 3). Real instruments are those producing a meaningful measurement for a variable, obtained either by a human observation (e.g. visibility) or by a measuring device (e.g. rainfall, temperature, etc.). Derived instruments do not perform any real measurements; their "measurements" are the result of calculations/transformations on other data. For example, discharge in an hourly or daily time scale corresponds to a derived instrument, as it is derived from a stage-discharge diagram. The latter is also a derived instrument with "measurements" produced from stage and direct discharge measurements (Figure 3).

Time series data can be classified as raw or aggregated (Nalbantis & Tsimbides, 1993). Raw data are the direct or processed measurement outcomes of the instruments, real or derived. Such data form a time series having some time resolution (time step), e.g. hourly, daily, monthly, annual or even irregular. The time resolution is imposed by the frequency of the measurements. Aggregated data are produced when an aggregation function, which effectively broadens the time step (or reduces the time resolution), is applied to raw or even other aggregated data. While raw data may have any time step, aggregated data are stored in the database only in the standard time steps, i.e., hourly, daily, monthly and annual. Other time resolutions may be produced on-line by the user, by applying the aggregation function to the proper data sets.

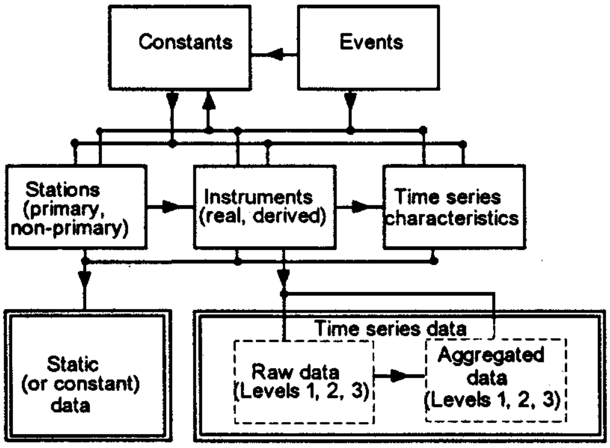
Constants
Events
Stations (primary, non-primary)
Instruments (real, derived)
Time series characteristics
Static (or constant) data
Time series data
Raw data (Levels 1, 2, 3)
Aggregated data (Levels 1, 2, 3)

Figure 2: Data and Metadata objects (in boxes with double and single lines, respectively) and their connections (arrows) and classifications (in brackets)

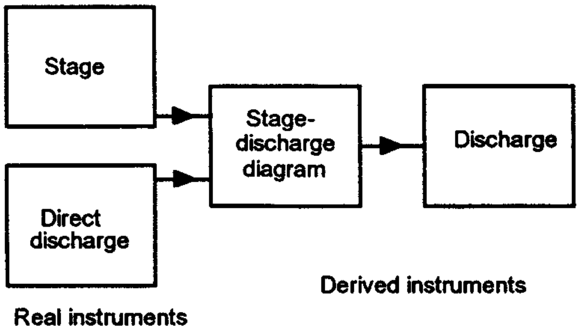

Figure 3: An example of real and derived instruments.

Raw data are subsequently classified in three levels, depending on the type of processing that has been applied to them (Figure 4):

1. Level-1 data are the original data, i.e. the result of the instrument measurements, together with background conditions, i.e. snow, ice, etc.

2. Level-2 data are checked data. The checks are applied to any single data value and include temporal, spatial and internal consistency checks, range checks, etc. Also, corrections to single erroneous values are included. The resulting data are characterised according to their quality and reliability.

3. Level-3 data are the result of infilling operations on time series with missing data, using various techniques and tools, depending on the type of the missing data and the availability of reference data. The techniques used for infilling include: mean value, reciprocal distance, normal ratio (Foufoula, 1983), simple and multiple linear regression (Matalas & Jacobs, 1964; Fiering, 1963), and univariate or multivariate first-order autoregressive models (AR(1); Salas, 1992).

The aggregated data are also classified in the same three levels. The data of a given level can be produced either from raw or aggregated data (with a higher time resolution) of the same level or from aggregated data (with same time resolution) of the previous level (Figure 4).

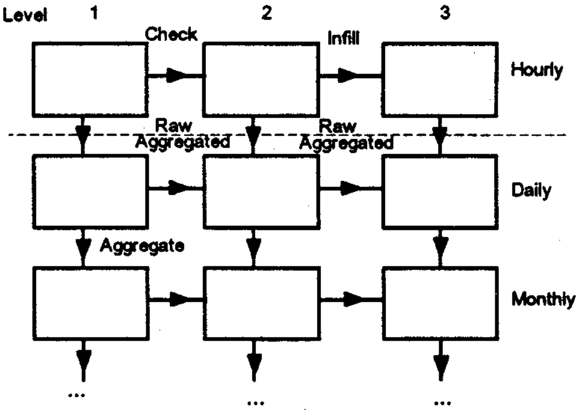

Figure 4: Explanation sketch for the two time series data classifications – raw and aggregated data and their three levels.

GENERAL DATABASE DESIGN

The database design is quite straightforward. Separate tables exist for metadata objects, each row of these tables referring to one object. For example, the stations table contains rows, each row holding the information on one station. Specially formatted integers (HYDROSCOPE id's) uniquely identify these objects. Redundant information on metadata is used to enhance performance whenever possible, to the expense of some disk space. Words instead of plain codes are used for the types of the objects, to enhance user readability.

Each of the participating services maintains its own stations and instruments, and measuring procedures. Every station may also contain various instruments, depending on the service, the variables of interest, the geographical area, etc. Therefore, there is no common measuring standard, format, time step or checking method among the services or even within the same service. Thus, time series data can be identified only by the instrument and the date to which they refer and are stored in the following general standard storage structure:

```
record = instrument_id date status
         <value>
```

where <value> is the measured, derived or aggregated value, instrument_id is the instrument to which the <value> refers, date is the respective date, and status is the status word.

The status word is a 32-bit (4-byte) integer, whose bits describe the characteristics and the state of the particular record and its <value>: the checks and infilling operations that have been applied to it, its level, the background conditions during the measurement (for raw data), etc. It must be noted that, when the <value> is acceptable to more than one levels, the same physical database record is used; some status bits are just "turned-on" to indicate that the record belongs to the respective levels.

The <value> need not be a single value. In general, it consists of one or more parts, like:

```
<value> = <part0>[<part1>...<partn>]
```

This is because not all instruments measure in the same way. For example, a rainfall measuring instrument measures only the rainfall depth; a wind measuring instrument measures both the wind speed and the wind direction; a soil temperature instrument measures the soil temperature for six different depths. Thus, the corresponding <value> field consists of one, two and six parts, respectively. Each part is stored as an integer of a given width (1-, 2- or 4-bytes) and precision, although the real value is a 4-byte floating point number. The integer type has been chosen in order to reduce the disk storage and network bandwidth requirements. Given the measurement unit and the required precision (in decimal digits), the following relationship holds for each part:

<part> = real_part_value * 10^{precision}

To obtain the real value an application has to divide by 10^{precision}, and this is the cost of this methodology. For instance, temperatures and rainfall depths are stored as 2-byte integers with a precision of one decimal unit. Since two-byte integers have a maximum value of 32768, a temperature (or rainfall depth) measurement may have a value of up to 32768 / 10 = 3276.8 degrees (or mm). The format used to store the values of a specific time series is stored in the table containing the time series characteristics.

All values are "nullable", i.e. they may have the special "non-value" null that is used to denote the absence of a measurement. Thus, for any missing value (e.g. due to temporary instrument malfunction) the corresponding record must exist in the time series table, but all (or just the missing) parts are set to null. It must be noted however, that because of INGRES restrictions the cost paid to implement this property is high, i.e. one extra byte for each part of the <value>.

An alternative storage structure is the list structure. It is appropriate to store a series of <value> fields with a constant time step. The list storage structure may be used to store automated instrument readings, e.g. a full day of 24 hourly measurements for rainfall or wind, or a well-defined set of aggregated values, e.g. a full month of 30 or 31 daily values. The general layout of this format is:

```
list_record = instrument_id date status
              num_values time_interval
              <value0> <value1> ...
              value31>
```

for up to 32 values. The date field stores the starting date for the list that contains num_values elements. The ith <value> field refers to a date equal to date + (i-1) • time_interval. The <value> fields in the list storage structure have also formats and parts, i.e. they may actually compose of more than one parts, their width depending on the type of the variable.

Referential integrity rules (Date 1981) have been employed to ensure consistency of the database, e.g. that any instrument belong to one and only one station, etc. Powerful security features are built into the database (Pipili & Papakostas 1992). The scheme used is based on user groups that may or may not be permitted to perform certain operations (select, insert and update - delete) on certain categories of data (raw - aggregated, metadata). The system also permits accounting for any retrieved information. The accounting is based on the type of the data fetched, the number of rows and the processing time of the query.